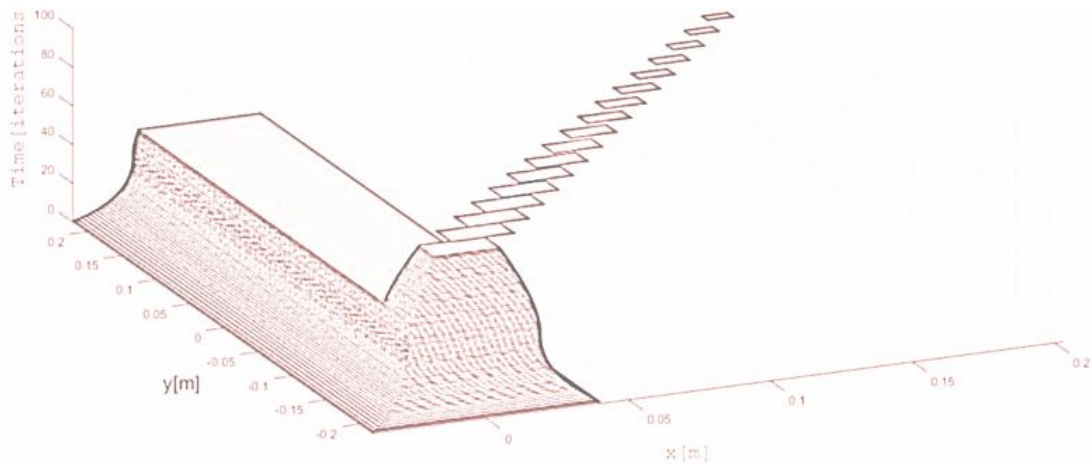
Robust MPC humanoid gait generation: a constraint restriction approach.

A.T.Elhelow
A.H.Abdalla
F.M.Smaldone

Sapienza Università di Roma
Autonomous and Mobile Robotics & Robotics 2 shared project
Project Assignment No.1
Academic year 2017/2018

February 14, 2020



Abstract

The technique of Tube-MPC has been explored in its general form and then applied to the case of the humanoid gait generation. A LQR stabilizer plus MPC have been used to consistently perform the technique. The case of intrinsically stable humanoid gait generation via boundedness equality constraint for the Tube-MPC has eventually been considered.

Contents

1 Introduction

Model predictive control has turned out to be a useful tool for humanoid gait generation by allowing an efficient constraint handling. The optimization-based technique is utilized so as to generate, at each iteration, the optimal gait trajectory of the system satisfying the given constraints. However, it is well known that the action of a bounded persistent disturbance can destabilize a predictive controller which has been designed to be stabilizing for the nominal case.

For example any parameters' uncertainty can become destabilizing for the gait and can be seen as a disturbance for the MPC algorithm[1]. Exploring and implementing a procedure (described by Chisci in [1]) able to counteract both of the possible sources of undesired behaviour (external disturbances or parameter uncertainties) in humanoids is the objective of this work.

The case study over which the disturbance rejection will be applied is the humanoid gait generation in the case of preassigned footsteps over an ideal plan terrain. Under the latter assumption, the ZMP coincides with the center of pressure and thus it becomes easily identifiable. The humanoid model for MPC is in fact chosen to be the linear inverted pendulum plus dynamic extension of [2] where the ZMP velocity is the input.

The work will follow this outline. A paper [1], where the Tube MPC is introduced, will be explained

[1]An inaccurate or uncertain model could be critical for the MPC which is strongly based on the knowledge of the model of the system.

in detail and a numerical example will be provided to simulate and appreciate the validity of the procedure. After a short discussion over disturbance modelling for humanoid LIP model, the problem of the Robust MPC gait generation will be faced and a solution, as close as possible to the approach of [1], will be provided with numerical simulations. To conclude, the issue of the boundedness equality constraint (introduced in [3]) is considered and an approach to use [1] for robustness and disturbance rejection is proposed and verified by a numerical simulation. The simulation platform will be Matlab, making also use of the MPT toolbox in order to consistently handle the constraints, as described in [1].

2 Systems with persistent disturbances: predictive control with restricted constraints

2.1 MPC framework

We introduce here the model predictive control in the specific settings of [1]. Consider the implicit, discrete, linear time invariant state space system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Dw(k) \\ y(k) &= Cx(k) \end{aligned} \tag{1}$$

subject to the state, output and control constraints $x(k) \in \mathbb{X} \subseteq \mathbb{R}^n$, $y(k) \in \mathbb{Y} \subseteq \mathbb{R}^q$ and $u(k) \in \mathbb{U} \subseteq \mathbb{R}^m$, in presence of the input disturbance $w(k) \in \mathbb{W} \subset \mathbb{R}^p$. Assume, for convenience, that the pair (A, B) is stabilizable and that $\mathbb{U}$, $\mathbb{W}$ and $\mathbb{Y}$ are compact sets.

Model Predictive Control will be used to design a non-linear state feedback controller

$$u(k) = g(x(k)), \tag{2}$$

such that the system is stable and satisfies the constraints.

Consider (1) and assume the evolution of the system is regulated in a small neighborhood of the origin by a linear stabilizing static state feedback controller of the form

$$u = Fx(k). \tag{3}$$

Since the system is subjected to state, output and input constraints, a stabilizer which regulates the states to the origin is not guaranteed to let the system satisfy its constraints. Assume, also, that an MPC algorithm is also running to this purpose (constraints satisfaction), so that the closed loop state space equations in presence of both controllers can be expressed as

$$x(k+1) = Ax(k) + Bg(x(k)) + Dw(k) \tag{4}$$

which can be rewritten as

$$x(k+1) = \Phi x(k) + Bc(k) + Dw(k) \tag{5}$$

where $\Phi \triangleq A + BF$ and $c(k) = [g(x(k)) - Fx(k)]$ is the MPC resulting control variable. According to the well known MPC paradigm, at each iteration step the following optimization problem (see Section 6) is solved for N (prediction horizon) future control moves:

$$c = \arg\min_{c} J(c) \tag{6}$$

$$J(c) = \sum_{n=0}^{N-1} c^T(k+n|k)\Psi c(k+n|k) \qquad \Psi = \Psi^T > 0, \tag{7}$$

where $c = [c(k|k), c(k+1|k), \dots, c(k+N|k)]$, -the cost function can be modified according to the specific needs- under the constraints

$$\begin{aligned} x(k|k) &= x(k) \\ x(k+n+1) &= \Phi x(k+n|k) + Bc(k+n|t) \qquad n \geq 0 \\ x(k+n|k) &\in \mathbb{X} \\ y(k+n|k) &\in \mathbb{Y} \\ u(k+n|k) &\in \mathbb{U}. \end{aligned} \tag{8}$$

Take then $c(k) = \hat{c}(k|k)$, first value of the optimization solution, and apply to the physical system:

$$u(k) = Fx(k) + c(k). \tag{9}$$

It is clear that if the origin is contained in $\mathbb{X}$, $\mathbb{Y}$ and $\mathbb{U}$, the sought nonlinear controller $u = g(x(k))$ tends, for $t \to \infty$ to $u = Fx(k)$ in case of absence of external disturbances.

The presence of persistent disturbances acting on the system may render the MPC and the stabilizer not sufficient anymore to guarantee the constraints satisfaction.

2.2 Robust Model Predictive Control with constraints restriction

This subsection will show a method to modify the constraints so that any MPC solution which satisfies the modified constraints would also satisfy the original constraints, even in presence of a persistent disturbance. The general idea relies on a constraints restriction which, in case of stability of the system (boundedness of the state is at least required), guarantees boundedness of the state response in presence of disturbances and satisfaction of the original constraints. By a specific choice of the restriction, the state will be forced to converge to a disturbance invariant set i.e. an invariant set with respect to the disturbance action.

We formally introduce now the concept of d-invariance, over which the erosion procedure is based on:

Definition 2.1 *A set $\Sigma \in \mathbb{R}^n$ is said to be d-invariant if, $\forall x \in \mathbb{X}$ and $\forall w \in \mathbb{W}$, $Ax + Dw \in \Sigma$.*

Consider now the set of the reachable states by the action of the bounded disturbance, which, for the closed loop system (under the action of a stabilizer $u = Fx$), has the following form:

$$R_j = \sum_{n=0}^{j-1} \Phi^n Dw. \tag{10}$$

The series of the sets R_j has a convergence limit for $t \to \infty$ if the matrix Φ is asymptotically stable[2], and this being the case, R_∞ is a compact set.

Consider now the nominal MPC (6). This predictive algorithm is not supposed to take into account the action of persistent external disturbances, like a modeling error or parameters uncertainties. In order to obtain a satisfying behaviour in presence of exogenous undesired inputs, a constraints restriction (for the optimization problem), based on (2.1, 25), can be performed; a *sufficient* condition for robustness, i.e. disturbance rejection, is thus to replace (8) with new constraints defined in the following way:

$$\begin{aligned} x &\in \mathbb{X}_k \qquad \textit{where} \quad \mathbb{X}_k \triangleq \mathbb{X} \sim R_k \\ y &\in \mathbb{Y}_k \qquad \textit{where} \quad \mathbb{Y}_k \triangleq \mathbb{Y} \sim CR_k \\ u &\in \mathbb{U}_k \qquad \textit{where} \quad \mathbb{U}_k \triangleq \mathbb{U} \sim FR_k. \end{aligned} \tag{11}$$

The symbol "$\sim$" expresses the Pontryagin difference between the two sets (i.e. given two sets A and B, $A \sim B = \{a | a + B \subseteq A\}$). By this action, the constraints are eroded so that all the actions of acceptable persistent disturbances will be a priori counteracted by the optimization problem itself[3].

[2]This condition is guaranteed by the controller $U = Fx$.

[3]Another issue for the procedure are the initial conditions for the optimization to be solved: not all initial state values ensures a solution in case of constraints restriction.

Before giving a numerical example, it worth dedicating a few words on some remarks regarding this procedure. The existence of $\mathbb{X}_k$, $\mathbb{Y}_k$ and $\mathbb{U}_k$ is only guaranteed, according to the definition of Σ, if

$$R_j \subset \mathbb{X} \qquad j = 1, 2, \ldots, N. \tag{12}$$

This condition strongly affects the feasibility of the erosion procedure. As matter of fact, a linear state feedback controller F, so that the state matrix $A + BF$ is asymptotically stable, is a valid tool by means of which it's possible to externally ensure the existence of a limit (say R_∞) of the R_j series and the actual existence of a well identifiable minimal d-invariant subset. So the choice of the controller, if required by the system, may play a determinant role in the disturbance rejection procedure design since it might be possible to obtain, in case of "slow" convergence[4], $\mathbb{X} \subset R_j$ from a certain prediction step. In order to render the concept clearer, it's possible to look at the Figure 3 (that is derived from numerical simulations of the next subsection), where different static state feedback controllers are applied to the system and different restrictions are performed according to the controllers. An interesting observation, that will be exploited later on, is the fact that the above stated procedure is based on a *sufficient* condition [1]. The violation of condition (12) would only imply the impossibility to use a d-invariant subset as a target set from a certain prediction step: the disturbance rejection might however be possible.

2.3 Numerical example

A simulated example is shown in this subsection to highlight the validity of the procedure. Consider the system matrices (taken from [1]):

$$A = \begin{bmatrix} 1.1 & 1 \\ 0 & 1.3 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad ,$$

assume the system is subject to the following input and output constraints

$$\begin{aligned} \mathbb{U} &= [-1, 1] \\ \mathbb{Y} &= [0.05, 2] \end{aligned} \tag{13}$$

and the disturbances admissible set is

$$\mathbb{W} = \{w : ||W||_\infty \leq 0.2\}. \tag{14}$$

The prediction horizon is chosen to be $N = 20$ while the cost function is so chosen

$$J(c(k)) = \sum_{n=0}^{N-1} c^T(k+n|k)\Psi c(k+n|k). \tag{15}$$

A LQR controller in synthesized[5]; the system is asymptotically stable and, from the initial conditions, the state trajectories would converge to the origin, but the output constraint of (13) would not be satisfied since the lower bound for y is supposed to be 0.05 and not 0. The presence of a disturbance within the acceptable set generates the evolution in Figure 1, over which the output is strongly violating its constraints. Constraint restrictions (i.e. a Pontryagin difference) is performed by means of the Matlab MPT toolbox [5],

[4] With $u = Fx$, it's possible to move the eigenvalues of the closed loop state matrix Φ so that the system converges to the origin in a faster or slower rate.

[5] The following weights were used: $Q = I_2$, $R = 0.01$. As suggested in [1], the weight Ψ for the cost function can be optimally obtained as $\Psi = R + B^T PB$, where P is the solution of the Riccati equation of entries A, B, Q and R.

which allows an easy polygon's handling. Another fact which has to be taken into account is that the input constraints are generally given for $u = Fx + c$ and not for c (the MPC minimization variable). So besides the polygonal Pontryagin difference, a reparametrization of the constraints is generally required. By means of the restriction, the effect of the disturbance with respect to the output and input constraints is rejected. The steady state behaviour of the output remains fixed to the value of 0.05, which is perfectly inside the given output constraints (Figure 2). The output constraints restriction is shown in Figure 3 in order to give a graphical idea of the procedure.

2.4 Figures

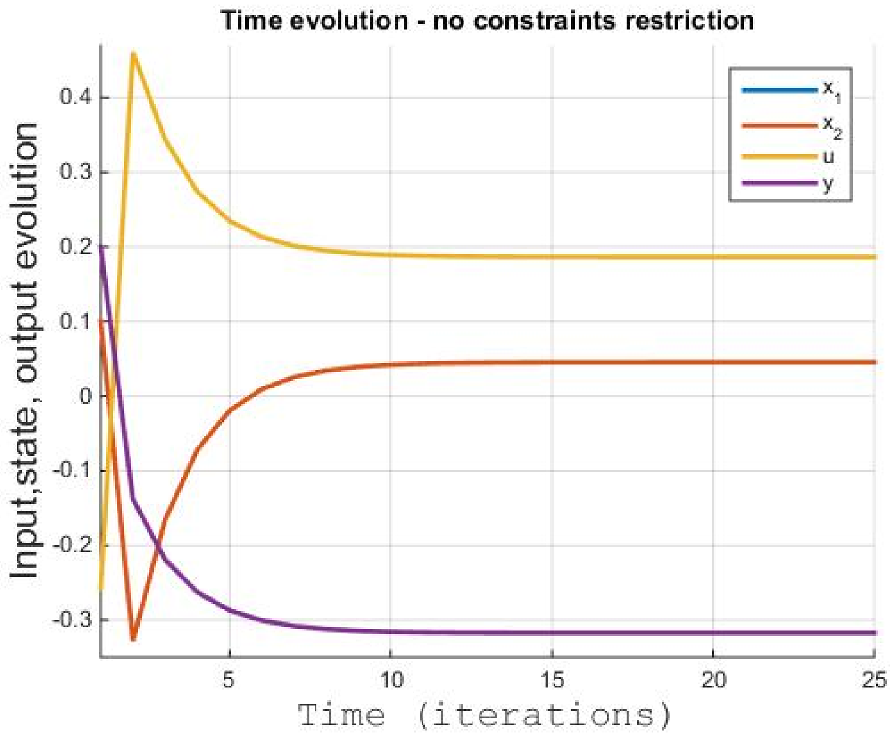


Figure 1: Closed loop time evolution of the system with disturbance. The states and the output assume the steady state behaviour that the exogenous disturbance impose, completely violating the output constraints.

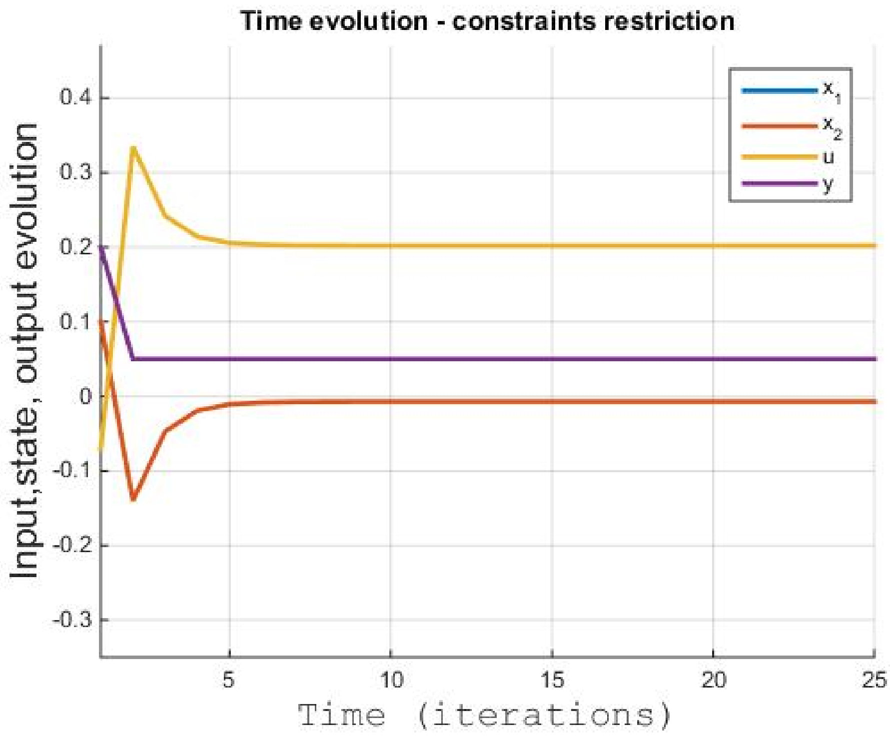


Figure 2: Closed loop time evolution of the system with disturbance and constraint restriction. The states and the output assume the steady state behaviour that the exogenous disturbance impose, but, by means of the restriction, the given constraints are no more violated.

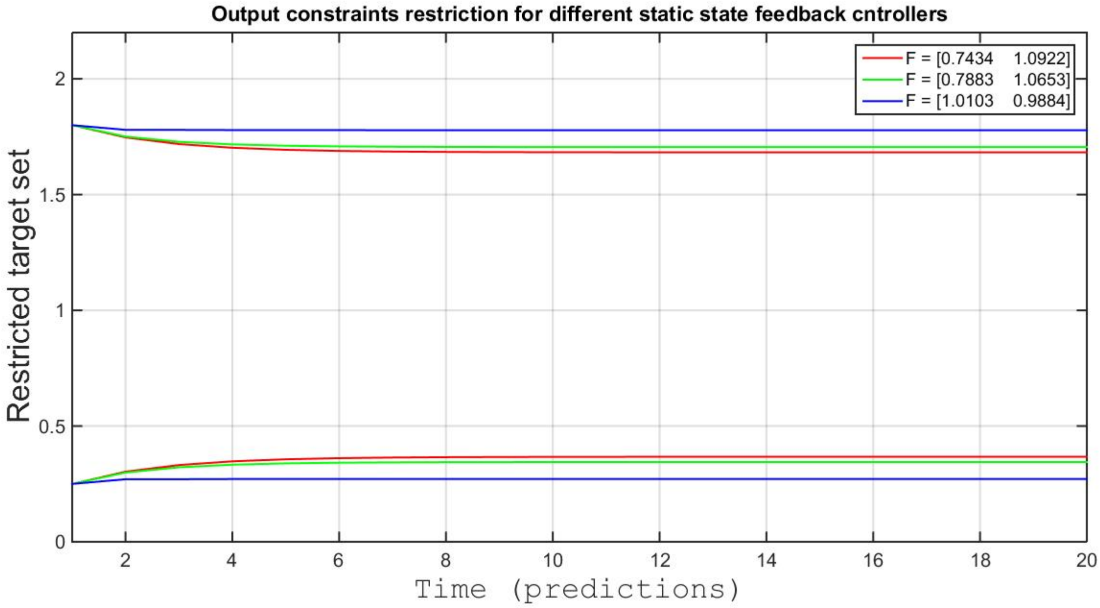


Figure 3: The output constraint restriction, after the "erosion", is represented by this picture for various LQR controllers (obtained by varying Q, R). Since there exists a limit for R_j, previously called R_∞, the entity of restriction is limited in time.

3 Humanoid gait generation via MPC in case of persistent disturbances

3.1 Constraints restriction for humanoid gait generation

The aim of this section is to apply the constraint restriction procedure for persistent disturbance rejection to the MPC for humanoid gait generation in case of preassigned footsteps. A linear model for the MPC is required and an effective one which is efficiently able to describe the humanoid center of mass and ZMP behaviour is the LIP + dynamic extension model i.e.

$$\begin{bmatrix} \dot{x}_c \\ \ddot{x}_c \\ \dot{z}_z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \omega^2 & 0 & -\omega^2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ \dot{x}_c \\ z_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{x}_z, \tag{16}$$

where $\omega = \sqrt{g/h_{CoM}}$ and h_{CoM} is the constant height of the CoM.

Two issues have to be taken into account in the gait generation problem. According to the LIP model (16), an unstable mode is present[6] and, without properly considering it, a stable gait is not obtainable by the humanoid. Furthermore the model stability doesn't imply dynamic balance for the gait; a sufficient condition to guarantee it is that the zero moment point remains inside the support polygon: an acceptable CoM trajectory would be generated in this way, as a consequence of the condition.

In order to directly apply the constraints restriction procedure of the previous section, since the A state matrix is unstable, we choose to design a linear time invariant stabilizing state controller of the form (3) for the model of the system to be controlled. The MPC will then be used to force the ZMP to lie inside the support polygon and to ensure this behaviour even in presence of undesired disturbances. The application of the method [1] to the humanoid gait generation yields to a more complex perspective of the problem. The two levels of stability (i.e. boundedness of state evolution and dynamic balance) are strongly connected: a necessary condition to obtain a satisfying ZMP trajectory is that the evolution of the states are bounded, and the input integral (ZMP) is subjected to the support polygon constraints; any controller will have to deal with them. The gait generation problem can be thus seen as a critically constrained system stabilization.

The ideal objective would be to find a piecewise constant function $u(k)$, i.e. a ZMP velocity command, to be applied to the physical humanoid in order to obtain a stable and dynamically balanced gait in spite of the action of undesired disturbances.

3.2 Numerical examples

We show in this subsection a numerical simulation of the procedure. The model parameter ω of (16) has the nominal value given by the parameter's choice $h = 0.8\ m$ and of course $g = 9.81\ m/s^2$. Suppose an external

[6] In fact, in the LIP model

$$\ddot{x}_c = \omega^2(x_c - x_z)$$

it's possible to separate the stable and unstable modes :

$$x_s = x_c - \dot{x_c}/\omega$$

$$x_u = x_c + \dot{x_c}/\omega.$$

destabilizing disturbance[7] with the following form is acting on the system:

$$D = \begin{bmatrix} 0 \\ 0.01 \\ 0 \end{bmatrix} \tag{19}$$

and the amplitude[8] w_d is assumed to be unknown but bounded in the set $[-0.12, 0.12]$. The system is stabilized by a LQR static state feedback controller $u = Fx(k)$ and thus the model to which apply the MPC is formally the same as (5). The resulting MPC optimization problem will be the same as (15), with $\Psi = 10$ and of course, as optimization constraints, the support polygon with reparametrized control variable (see Section 6). From Figure 5, it's immediate to see how the restriction procedure is able to counteract the presence of the disturbance. The effect of the erosion on the constraints is illustrated in Figure 4.

We have also considered the case of parametric uncertainties. In the adopted LIP model the only source of uncertainties can be the height of the center of mass which can be see as an uncertainty on the ω parameter of the model ($\omega = \sqrt{g/h_{CoM}}$). Assume in fact that the value of ω is affected by a 10% uncertainty around its nominal value. Keeping in mind the simplest error propagation formula[9], the relation between $\Delta\omega$ and Δh is the following

$$\Delta\omega = \frac{g\Delta h}{2h^2}\sqrt{\frac{h}{g}} \tag{20}$$

which means that the original relative uncertainty of h was 4.6%. Assume now that the model uncertainties can be modeled as the action of an exogenous bounded disturbance (as in (1)). Consider the state matrix in presence of parametric uncertainties (20):

$$\begin{bmatrix} 0 & 1 & 0 \\ (\omega^* + \Delta\omega)^2 & 0 & -(\omega^* + \Delta\omega)^2 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \omega^{*2} & 0 & -\omega^{*2} \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \Delta\omega^2 + 2\omega^*\Delta\omega & 0 & -\Delta\omega^2 - 2\omega^*\Delta\omega \\ 0 & 0 & 0 \end{bmatrix} \tag{21}$$

with $\Delta\omega \in [\Delta\omega_m, \Delta\omega_M]$ and ω^* as the true value. One may ideally represent the uncertain part of (21) as a persistent disturbance in this way

$$\begin{bmatrix} 0 & 0 & 0 \\ \Delta\omega^2 + 2\omega^*\Delta\omega & 0 & -\Delta\omega^2 - 2\omega^*\Delta\omega \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} |x_{c_{max}}| \\ |\dot{x}_{c_{max}}| \\ |x_{z_{max}}| \end{bmatrix} = \begin{bmatrix} 0 \\ |x_{c_{max}}| - |x_{z_{max}}| \\ 0 \end{bmatrix} (\Delta\omega^2 + 2\omega^*\Delta\omega) = Dw, \tag{22}$$

[7]Any external force acting on the CoM can be assumed to be source of disturbances , with the general shape

$$Dw = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} w. \tag{17}$$

Furthermore, a disturbance on the dynamic extension might be considered in the D column vector (and it has been done in simulations), yielding to

$$D = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}. \tag{18}$$

[8]In discrete domain.

[9]Since

$$\omega = f(h)$$

it's possible to expand it around it's true value by means of a taylor expansion truncated at the first order:

$$\omega \simeq \omega^* + \frac{\partial\omega}{\partial h}\Delta h,$$

where ω^* is the true value.

so that the implicit form of the model becomes:

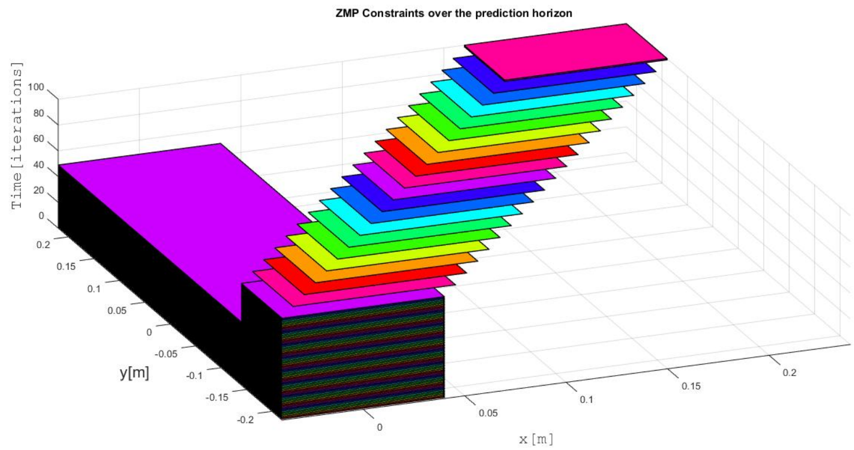
$$\begin{bmatrix} \dot{x}_c \\ \ddot{x}_c \\ \dot{z}_z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \omega^2 & 0 & -\omega^2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ x_c \\ z_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{x}_z + Dw, \tag{23}$$

where the $|x_{c_{max}}|$, $|\dot{x}_{c_{max}}|$ and $|x_{z_{max}}|$ are the maximal value obtained during simulations without disturbances[10]. By (22), it's easy to understand that a parametric uncertainty in a state space model parameter can behave as a an external force pushing away the center of mass.

There are many well-known robust control design methods in the literature, but in this specific situation, the tube MPC turns out to be a very effective tool. The reason relies upon the constraints under which the LIP model has to be forced: the input of the system is the ZMP velocity while the ZMP position has to lay within a specific support polygon. Tube MPC can guarantee robust stabilization while taking the constraints into consideration. By using the same LQR controller of the previous example and keeping in mind (22), we obtain the Tube-MPC behaviour of Figure 6.

[10] All these manipulations are required since the problem's formulation of [1] assumes the disturbance to be persistent in time (almost like a step disturbance with unknown, but bounded within certain constraints, amplitude acting on the system).

3.3 Figures



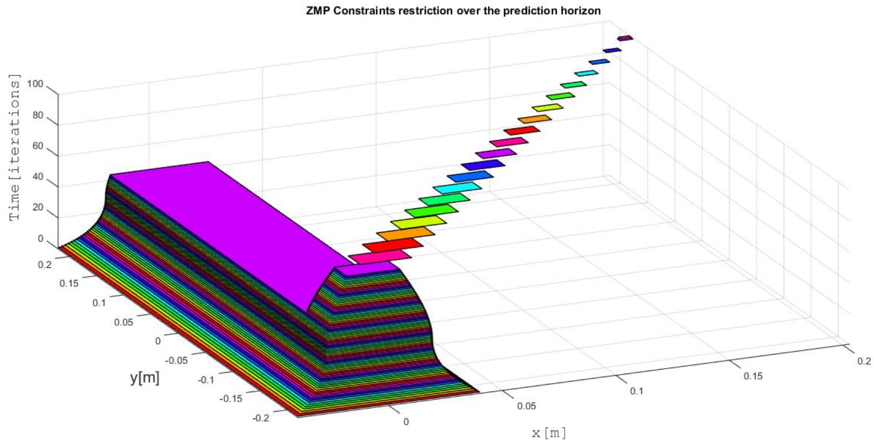
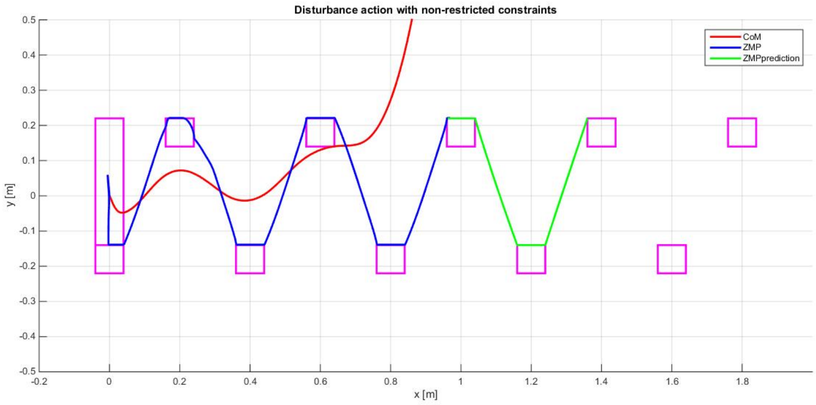


Figure 4: Graphical representation of the ZMP constraints before and after the restriction. After the restriction (the lower figure), the MPC will take, as a target set, the minimal d-invariant subset: if there exists a MPC solution satisfying these restricted constraints, the disturbance action will be rejected.



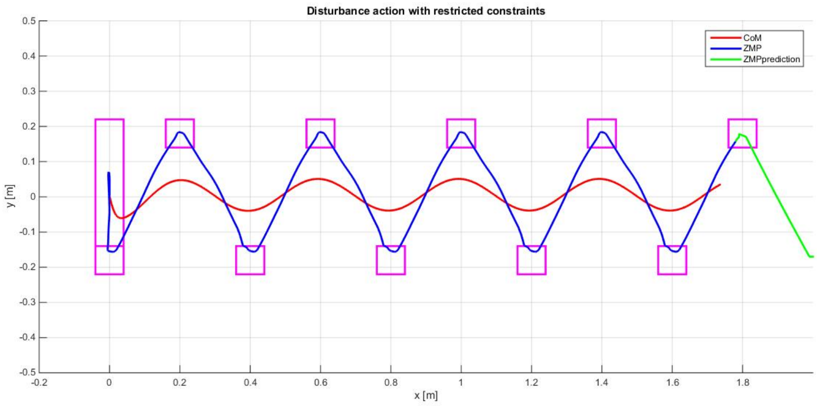
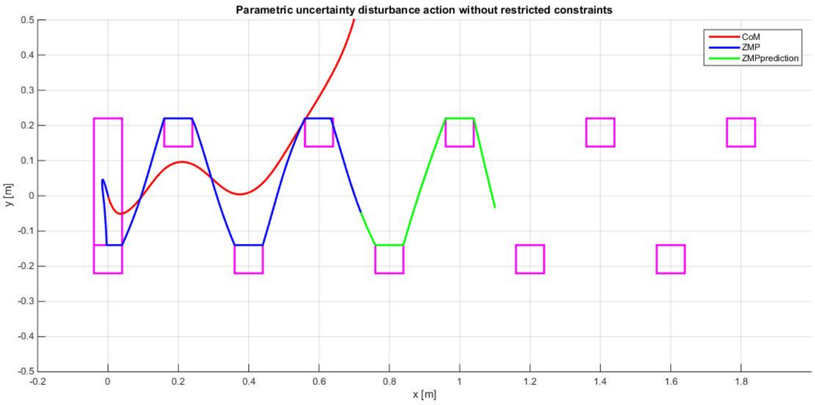


Figure 5: ZMP and CoM trajectories in case of a disturbance action. First picture shows how the action of this undesired disturbances is destabilizing for the system if no countermeasures are applied, while the second figure confirms the validity of the Tube-MPC algorithm.



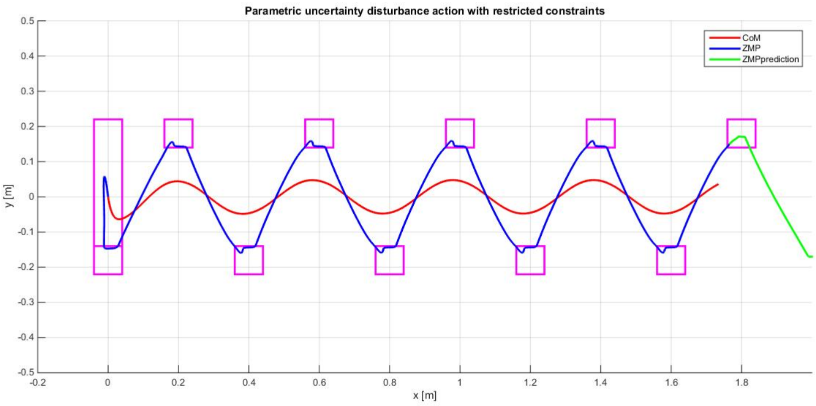


Figure 6: ZMP and CoM trajectories in case of parametric uncertainties. First picture shows how the action of this uncertainty may become destabilizing for the system (e.g. worst case value), while the second figure confirms the validity of the Tube-MPC algorithm.

4 Regarding the boundedness condition

4.1 Constraints restriction with boundedness condition

In [2], MPC is used to find a piece-wise constant control able to achieve dynamic balance for the gait and trajectories evolution boundedness for the linear inverted pendulum model. While exploiting the nature of MPC to maintain the ZMP inside the support polygon, a boundedness condition (represented by an equality constraint for the MPC optimization) is needed to cancel the unstable state evolution which would have led to instability. In particular, it is shown in [3], that by choosing the following initial condition

$$x_u(t_0) = \int_{t_0}^{\infty} e^{-\omega(\tau - t_0)} x_z(\tau) d\tau, \tag{24}$$

the LIP system has bounded evolution. This means that system (1) has directly as input $u(k)$ the solution of the MPC optimization, and the action of a static state feedback stabilizer is no more needed. The difference from the original method of [1] relies on the capability of the MPC to both ensure system boundedness via the equality constraints, and ensure dynamic balance by forcing the ZMP to be inside the support polygon.

We attempt here to apply the constraint restriction procedure for persistent disturbance rejection to the above mentioned intrinsically stable MPC for humanoid gait generation.

Note that the expression of R_j, namely the set of reachable states by the action of a bounded disturbance, with the omission of the state feedback controller (3) becomes instead

$$R_j = \sum_{n=0}^{j-1} A^n D w. \tag{25}$$

It is important to note here that since the boundedness of the system is now guaranteed by the equality constraint of the MPC, and with the intrinsic instability of the LIP model system matrix A, it would be highly probable that a set R_j becomes larger than the state constraint $\mathbb{X}$. It means that no minimal d-invariant subset exists (so no more non-empty d-invariant subset can be found). Although $\mathbb{X}$ becoming empty might violate a sufficient condition of the method in [1], we show that disturbance rejection in the gait generation might still be achieved. We introduce now an algorithm to face this issue.

Assume that for some disturbance w, there exists, for some $J = 1, 2, ..., h$ future predictions, where $h \leq N$, h non-empty d-invariant sets $\mathbb{X}_k$ and the remaining $N-h$ $\mathbb{X}_k$ are empty. If one chooses the constraint restriction for the MPC constraints over the prediction horizon as

$$\mathbb{X}_1, \mathbb{X}_2, ..., \mathbb{X}_h, \mathbb{X}_h, ..., \mathbb{X}_h$$

i.e. replace the empty d-invariant target sets, generated by the erosion from $h+1$ to N, with the smallest existing d-invariant set $\mathbb{X}_h$, and if a solution could be found by the MPC, then the disturbance w would surely be rejected correctly by the system for the first h future states. If h is large enough, and since the MPC only applies the computed control for the first prediction step, the stability of the system for the whole gait with the applied disturbance could be achieved. Note that even though no disturbance rejection could be achieved after h steps, if a solution could be found by MPC, disturbance w would surely be rejected correctly by the system for the first h future states. The choice of replacing empty $\mathbb{X}_k$ with $\mathbb{X}_h$ might be giving a sense of the future nature/direction of the gait trajectory. This, with the fact the MPC only applies the computed control for the first prediction step, might explain the capability of the MPC to correctly achieve robust stability of the system for the whole gait.

Figure 7 gives a graphical explaination of the previous considerations. Note that we found that this procedure works for disturbances that result in an empty $\mathbb{X}$ as early as in the second prediction step, i.e. $h = 1$. This will be further illustrated in the next section.

4.2 Numerical example

Consider the same D column matrix as (18) and assume the persistent disturbance signal is bounded within the set $[-0.133, 0.133]$. The system is subjected to boundedness equality constraints in addition to the support polygon constraints for the ZMP position. With such a disturbance set the existence of a d-invariant subset is ensured only for the first 30 predictions. From that point on we choose as a target set $\mathbb{X}_{30}$ until the N_{th} prediction. The MPC cost function is chosen to be the same as (15), subjected to support polygon and bounded equality constraints (so as in [2]); graphical results of the constraints restriction are shown in Figure 8. The MPC is able to compensate the action of a disturbance even if the target set is not guaranteed to be a d-invariant subset: the intrinsically robust nature of the MPC seems to be able to recover the eventual errors in the compensation by recomputing a new solution at each iteration.

4.3 Figures

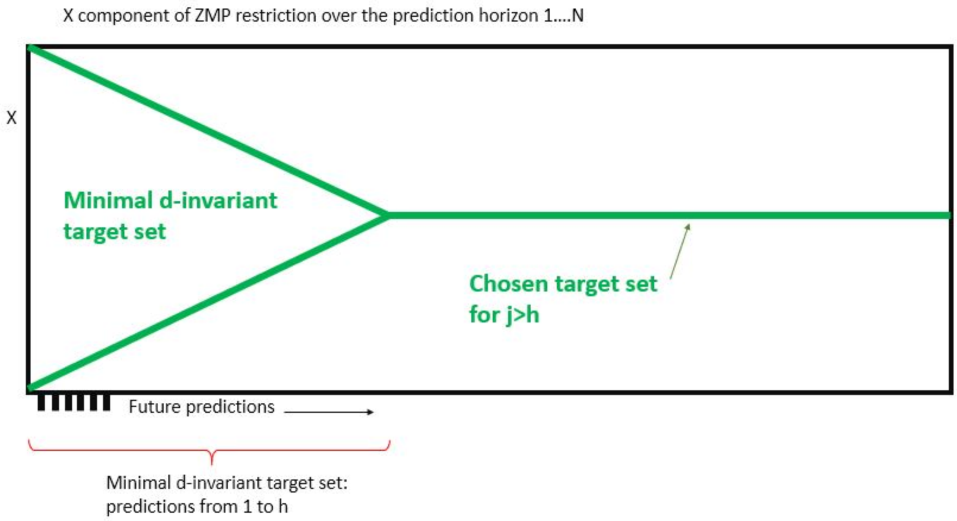
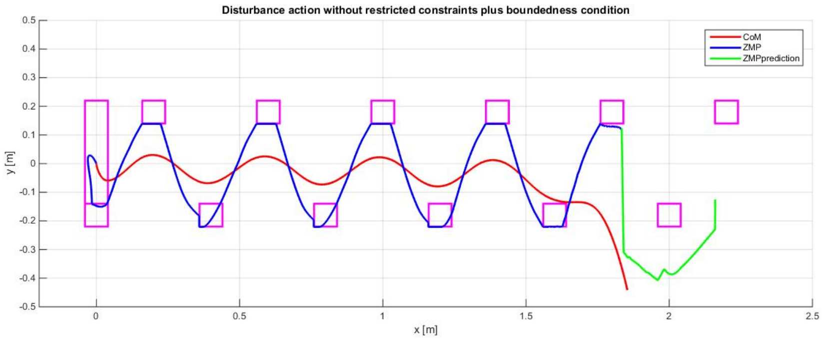


Figure 7: Evolution of d-invariant subset. From h on the target set is chosen to be $\mathbb{X}_h$. The solution, if it exists, would lay into this set and thus, the action of a non compensated disturbance, which would start being effective from h step on, would not bring the ZMP outside the support polygon since a new MPC solution will be computed at the next time instant.



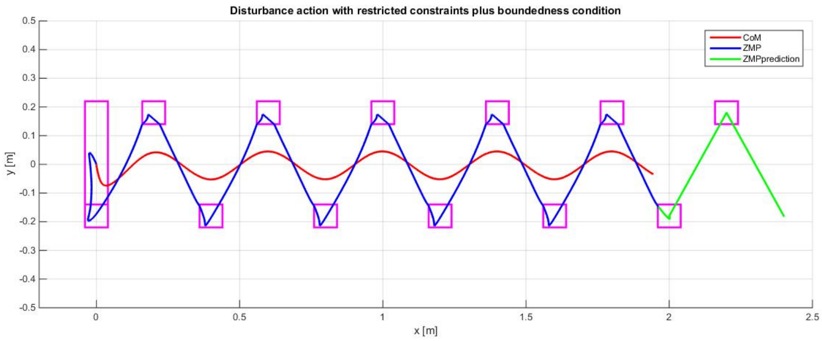


Figure 8: ZMP and CoM trajectories in case of disturbance action when the MPC has an equality constraint which is responsible for state boundedness. Although the existence of a d-invariant subset to be chosen as a target set for the algorithm is ensured for only 30 predictions (since the A matrix is unstable), with the proposed constraints restriction the MPC is able to counteract any undesired action of the disturbance.

5 Conclusions

Tube MPC has been confirmed to be a valid tool for disturbance rejection and for parametric uncertainties robustness. The constraints erosion so that the state evolution of the system approaches a d-invariant subset is able to reject a variety of destabilizing disturbances in a cost-effective fashion: a complex problem, under both a control and physical point of view, is transformed into a simple computational geometry manipulation. Although theoretically valid, an experimental validation should be performed to correctly understand how to apply the restriction procedure to a real-time physical system. The gait generation we have considered is only related to preassigned footsteps onto a flat ground. Online constraints erosion of unknown future footsteps may need to be investigated since a direct application of this work could not efficiently fit for the problem.

6 Appendix

We report, for completeness, all the computations we have developed in order to apply the optimization algorithm in the MPC.

6.1 MPC quadratic programming: computations

Given a cost function in the general discrete time form

$$J(c(k)) = \sum_{n=0}^{N-1} u^T(k+n|k)\Psi u(k+n|k) + \sum_{n=0}^{N-1} x^T(k+n|k)Qx(k+n|k) \tag{26}$$

with $\Psi = \Psi^T > 0$, $Q \geq 0$, in order to solve a quadratic programming problem with MATLAB 'quadprog' function, it's necessary to manipulate it so that the well known form

$$J(u) = \frac{1}{2}u^T Hu + \hat{F}^T u \tag{27}$$

is obtained (the constatant which derives from the computations is omitted since it doesn't affect the minimization solution). Keeping in mind that

$$x(k) = A^k x_0 + \sum_{i=0}^{k-1} A^i B u_{k-1-i}, \tag{28}$$

it's possible to rewrite

$$J(c) = u^T \Psi u + [A^k x_0 + \sum_{i=0}^{k-1} A^i B u_{k-1-i}]^T Q[A^k x_0 + \sum_{i=0}^{k-1} A^i B u_{k-1-i}], \tag{29}$$

from which

$$\begin{aligned} H_k &= 2[1 + (\sum_{i=0}^{k-1} A^i B)^T (\sum_{i=0}^{k-1} A^i B)] \\ \hat{F}_k^T &= [2(A^k x_0)^T (\sum_{i=0}^{k-1} A^i B)] \end{aligned} \tag{30}$$

for $k = 1, \ldots, N$. It's possible then to compact the (32) in a matrix form, so that all the k future instants will be considered into the same notation as in (27).

Define:

$$\bar{S} = \begin{bmatrix} B & 0 & 0 & \ldots \\ AB & B & 0 & \ldots \\ A^2B & AB & B & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ A^{N-1}B & A^{N-2}B & \ldots & \ldots \end{bmatrix}, \qquad \bar{T} = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \ldots \\ A^N \end{bmatrix}, \tag{31}$$

and of course

$$\begin{aligned} H &= 2[I + \bar{S}^T \bar{S}] \\ \hat{F}^T &= [2\bar{T}^T \bar{S}]. \end{aligned} \tag{32}$$

The inequality constraints wille be fed to the Matlab 'Quadprog' function in the same fashon:

$$Au \leq b. \tag{33}$$

Since the constrained variable is the ZMP which, according to (16), is the third state, we need a transformation to convert the ZMP constraints into input constraints. Two ways are possible (yielding the same result): the first, used in [2], simply converts the ZMP velocity (input) in ZMP position (constrained variable) by integration i.e. multiplying for the sampling time and then summing for each prediction. We have chosen, instead, a more general, but in this case heavier, representation.

By defining the ZMP as output via a so chosen C matrix:

$$C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \tag{34}$$

we are able to represent the constraints in the following manner

$$\begin{aligned} &\begin{bmatrix} CB & 0 & 0 & ... \\ CAB & CB & 0 & ... \\ CA^2B & CAB & CB & ... \\ ... & ... & ... & ... \\ CA^{N-1}B & CA^{N-2}B & ... & ... \end{bmatrix} \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ u(k+2|k) \\ ... \\ u(k+N-1|k) \end{bmatrix} \leq \\ &\leq \begin{bmatrix} y_{max}(k|k) \\ y_{max}(k+1|k) \\ y_{max}(k+2|k) \\ ... \\ y_{max}(k+N-1|k) \end{bmatrix} - \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ ... \\ CA^N \end{bmatrix}. \end{aligned} \tag{35}$$

An equivalent relation holds for y_{min}. The formulation is valid both for the x and for the y axis constraints.

6.2 Reparametrization: computations

In order to correctly address the LQR-MPC Robust Humanoid gait generation problem, we had to reparametrize the constraints according to the minimization variable $c(k)$ (as chosen in (7)). The ZMP constraints (35) are in fact adapted to be fed to the algorithm when u is the minimization variable, while, in the LQR-MPC approach, we make also use of a static state gain stabilizer. Considering (9), that we report for ease of reference

$$u(k) = Fx(k) + c(k), \tag{36}$$

we proceed with the following computations. By developing (9) for the prediction horizon, we have

$$\begin{bmatrix} u(k|k) \\ u(k+1|k) \\ u(k+2|k) \\ ... \\ u(k+N-1|k) \end{bmatrix} = F \begin{bmatrix} x(k|k) \\ x(k+1|k) \\ x(k+2|k) \\ ... \\ x(k+N-1|k) \end{bmatrix} + \begin{bmatrix} c(k|k) \\ c(k+1|k) \\ c(k+2|k) \\ ... \\ c(k+N-1|k) \end{bmatrix}; \tag{37}$$

and keeping in mind (5) (without considering the action of the disturbances), we have

$$\begin{aligned} \begin{bmatrix} x(k|k) \\ x(k+1|k) \\ x(k+2|k) \\ ... \\ x(k+N-1|k) \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 & ... \\ B & B & 0 & ... \\ \Phi B & B & 0 & ... \\ ... & ... & ... & ... \\ \Phi^{N-2}B & \Phi^{N-3}B & ... & ... \end{bmatrix} \begin{bmatrix} c(k|k) \\ c(k+1|k) \\ c(k+2|k) \\ ... \\ c(k+N-1|k) \end{bmatrix} + \\ &+ \begin{bmatrix} 0 \\ \Phi \\ \Phi^2 \\ ... \\ \Phi^{N-2} \end{bmatrix} x(k|k). \end{aligned} \tag{38}$$

Define now

$$\bar{S}_C = \begin{bmatrix} CB & 0 & 0 & ... \\ CAB & CB & 0 & ... \\ CA^2B & CAB & CB & ... \\ ... & ... & ... & ... \\ CA^{N-1}B & CA^{N-2}B & ... & ... \end{bmatrix}, \qquad \bar{T}_C = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ ... \\ CA^N \end{bmatrix}, \tag{39}$$

$$\bar{S}_F = \begin{bmatrix} 0 & 0 & 0 & ... \\ FB & B & 0 & ... \\ F\Phi B & FB & 0 & ... \\ ... & ... & ... & ... \\ F\Phi^{N-2}B & F\Phi^{N-3}B & ... & ... \end{bmatrix}, \qquad \bar{T}_F = \begin{bmatrix} 0 \\ F\Phi \\ F\Phi^2 \\ ... \\ F\Phi^{N-2} \end{bmatrix}, \tag{40}$$

and by plugging (38) into (37) and then (37) into (35), we get the final compact expression:

$$(\bar{S}_C\bar{S}_F)c \leq Y_{max} - (\bar{T}_C + \bar{S}_C\bar{T}_F)x_0. \tag{41}$$

which has exactly the same shape as (33). The constraints restriction has to be applied as presented in the previous sections directly to the boundary term, in this case Y_{max}.

References

[1] L. Chisci, J.A. Rossiter, G. Zappa, Systems with persistent disturbances: predictive control with restricted constraints, Automatica, Volume 37, Issue 7, 2001, Pages 1019-1028, ISSN 0005-1098;

[2] N. Scianca, M. Cognetti, D. De Simone, L. Lanari and G. Oriolo, "Intrinsically stable MPC for humanoid gait generation," 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, 2016, pp. 601-606;

[3] L. Lanari, S. Hutchinson and L. Marchionni, "Boundedness issues in planning of locomotion trajectories for biped robots," 2014 IEEE-RAS International Conference on Humanoid Robots, Madrid, 2014, pp. 951-958;

[4] A.Bemporad, "Model predictive control", lecture slides;

[5] MPT3 Wiki, toolbox documentation, https://www.mpt3.org/Geometry/Geometry.